Scorpio For the Developer

Posted At : May 2, 2007 10:09 AM | Posted By : Mark Kruger
Related Categories: Coldfusion's Future

No it is not the latest movie by Jean Claude Van Damme, nor is it a perfume by Este' Lauder. Scorpio is the code name for the next version of ColdFusion. At our meeting of the Nebraska Coldfusion User Group last night we heard a great presentation from ColdFusion specialist Adam Lehman previewing this enticing product. Needless to say it whetted our appetite. Here's *some* of the scoop.

Developer Stuff

On the developer side that had several innovative changes that really sound like they will make life easier for us.

- **CFSCRIPT Operators** - Scorpio will support the standard set of operators we are used to seeing from other languages like "++" and "!=". It looks like all of them will be supported *inside cfscript* and *most* of them will be supported in the tag-based logic as well. Obviously, operators that include angle brackets can't be supported inside of a tag (things like ">=") because of the nature of a tag based syntax.
- **Argumentcollection** - This attribute will now be supported inside of tags as well as functions. For example, an argumentcollection containing cache, datasource, blockfactor, username, password and name could be passed to the "cfquery" tag. You can override individual collection attributes as well, so you could have a "global" structure with all your dsn settings in it and just override the name. Pretty nifty.
- **CFIMAGE** - Coldfusion will expose the underlying Java stuff for working with images. Looking at a short list of possibilities it looks like things like captcha, watermarking, resizing, and converting will all be possible. Adam even said that the underlying system will be exposed in a set of function enabling it to be used as a "drawing API" - which kind of made me want to stab myself with my pen.
- **File IO** - Lots of file operations have been incorporated into the function library - things like reading a file lines or characters at a time, better directory operations etc. Again, simply exposing the underlying Java Architecture.

Client Side Code

This is one of the things I feel ambivalent about. CF has always been fairly poor when it comes to client side code. Its chief entry in this arena has always been CFFORM. Using CFFORM you can manage client side validation, create flash based forms, and implement all sorts of widgety client side behaviors. To my mind it always falls short because I can never seem to do *exactly* what I want. Instead I always seem to be able to do *mostly* what I want. We have always chosen to roll our own JavaScript and Ajax and allow CF to do what it does best - interact with databases and server systems. That's my full disclaimer.

Having said that I must say that there are some things here that intrigue me. For example:

- **Client Side CFCs** - Forgive me if I get this wrong, I'm doing it from memory. But using a new tag called CFAJAXPROXY you can instantiate CFCs on the client side and ColdFusion takes care of where the events and behaviors take place using AJAX. Under the hood, the CFC path is being passed to a JS function which (presumably) creates proxy functions and classes by unpacking the WSDL. The result is that the functionality of your CFC is exposed in your JavaScript. The implications? You can interact with your database and server from within your HTML page *without* a page refresh. In other words, the promise of Web 2.0 and Ajax but with a twist. Now your client side code can "mirror" your server side code - or replace it. It sounds interesting but the devil is in the details. For one thing, the example we saw had a dizzying array of libraries that needed to be included to make it work.

- **CFWINDOW and CFLAYOUT** - these 2 tags are widget tags. They create nice DHTML tabs and windows with lots of attributes attached to them for skinning and positioning.
- **PDF Stuff** - CF Document comes with TOC abilities and other capability for merging, extracting pages and the like. Actually some of these features might be a part of CFPDF (a new tag). CFPDF allows you to do one thing that I find interesting. You can create a PDF Form, upload it and extract the data from it. You can even put a submit button in it that pushes the form data to a CFM handler. A new "pdf" scope will contain all the fields from the form. This is a great solution for certain situations where having an "offline" form is important.
- **Rich Text Editor** - Scorpio will include an attribute for the CFTEXTAREA inside of CFFORM that will allow you to create a rich text editor automatically. Under the hood they are using FCKEditor - a popular open source editor that we have used many times.

Adam made the point that most of the client side enhancements are done by wrapping open source JS libraries with CF Tags. Several of the "web 2.0" items above are extensions of the Yahoo UI API. Indeed, they seem to be exposing much of the functionality of that UI in this product. The use of FCKEditor is another example. The thing that worries me is that you will now be dependent on Adobe for updates to your CF server to maintain browser compatibility. In other words, the server code base has a direct impact on how your content is viewed in the browser. That used to be the developers problem, but now it is Adobe's problem. Adam said they would be very aggressive about keeping these client side libraries up to date through the use of hot fixes. I'm excited to try some of these new tags - but it does worry me.

Tomorrow I will writ a bit more about the integration and management pieces of Scorpio. By the way, in the interest of full disclosure - I am *not* on the beta test for this product. The only thing I'm writing about is what I have heard in this one public meeting.